

CLAIM AMENDMENTS

This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Original) A method comprising the steps of:
receiving a first data stream of multimedia data, wherein the multimedia data includes a first protocol and further wherein the first protocol is unknown;
determining, based upon a first portion of the first data stream, the first protocol of the multimedia data.
2. (Currently Amended) The method as in Claim 1, wherein the first protocol is one of a set of predefined protocols comprising Motion Picture Experts Group 2 (MPEG-2), ~~Direct TV~~DIRECTV®, and Digital Versatile Disk (DVD) protocols.
3. (Original) The method as in Claim 1, further comprising:
storing a second portion of the first data stream in memory after the step of determining the first protocol.
4. (Original) The method as in Claim 3, wherein the second portion of the first data stream is received after the first portion of the first data stream.
5. (Original) The method as in Claim 3, wherein the second portion of the first data stream includes the first portion of the first data stream.
6. (Original) The method as in Claim 3, further comprising generating a database based on the second portion.
7. (Original) The method as in Claim 6, further comprising parsing the second portion of the first data stream to determine a first set of descriptors associated with the first data stream.

8. (Currently Amended) The method as in Claim 7, wherein the first set of descriptors includes a descriptor selected from the group consisting of~~from the set of descriptors comprising~~ a network identifier, multiplex information, and program information.

9. (Original) The method as in Claim 8, wherein multiplex information includes transport stream identifiers and program identifiers.

10. (Original) The method as in Claim 8, wherein the program information includes program numbers, program recovery clock identifiers, video data identifiers and audio data identifiers.

11. (Original) The method as in Claim 8, wherein the set of descriptors further includes elementary stream information and closed captioning information.

12. (Original) The method as in Claim 11, wherein the elementary stream information includes data stream types and elementary stream identifiers.

13. (Currently Amended) The method as in Claim ~~[[1]]~~3, wherein the memory includes a frame buffer.

14. (Original) The method as in Claim 1, further comprising:
 receiving a second data stream of multimedia data, different from the first data stream,
 wherein the multimedia data of the second data stream includes a second protocol,
 different from the first protocol and further wherein the second protocol is
 unknown;
 determining, based upon a first portion of the second data stream, the second protocol of
 the multimedia data of the second data stream.

15. (Cancelled)

16. (Cancelled)

17. (Cancelled)

18. (Cancelled)

19. (Cancelled)

20. (Cancelled)

21. (Cancelled)

22. (Cancelled).

23. (Cancelled)

24. (Cancelled)

25. (Cancelled)

26. (Cancelled)

27. (Cancelled)

28. (Cancelled)

29. (Cancelled)

30. (Cancelled)

31. (Cancelled)

32. (Cancelled)

33. (Cancelled)

34. (Cancelled)

35. (Cancelled)

36. (Cancelled)

37. (Cancelled)

38. (Cancelled)

39. (Cancelled)

40. (Cancelled)

41. (Cancelled)

42. (Cancelled)

43. (Cancelled)

44. (Cancelled)

45. (Cancelled)

46. (New) A device comprising,
a transport stream demultiplexor comprising:

an input configured to receive multimedia data, wherein the multimedia data
includes a first protocol and further wherein the first protocol is unknown;
a microcode engine configured to determine, based upon a first portion of the first
data stream, the first protocol of the multimedia data.

47. (New) The device of claim 46, wherein the device further comprises a memory configured to store a second portion of the first data stream after the microcode engine determines the first protocol.

48. (New) The device of Claim 47, wherein the second portion of the first data stream is received at the input after the first portion of the first data stream.

49. (New) The device of Claim 47, wherein the second portion of the first data stream includes the first portion of the first data stream.

50. (New) The device of Claim 47, wherein the device further comprises a stream engine coupled to an output of the transport stream demultiplexor, the stream engine configured to generate a database based on the second portion of the first data stream.

51. (New) The device of Claim 50, wherein the stream engine is further configured to parse the second portion of the first data stream to determine a first set of descriptors associated with the first data stream.

52. (New) The device of Claim 51, wherein the first set of descriptors includes a descriptor selected from the group consisting of a network identifier, multiplex information, and program information.

53. (New) The device of Claim 52, wherein multiplex information includes transport stream identifiers and program identifiers.

54. (New) The device of Claim 52, wherein the program information includes program numbers, program recovery clock identifiers, video data identifiers and audio data identifiers.

55. (New) The device of Claim 52, wherein the set of descriptors further includes elementary stream information and closed captioning information.

56. (New) The device of Claim 55, wherein the elementary stream information includes data stream types and elementary stream identifiers.

57. (New) The device of Claim 47, wherein the memory includes a frame buffer.

58. (New) The device of claim 46, wherein the input is configured to receive a second data stream of multimedia data, different from the first data stream, wherein the multimedia data of the second data stream includes a second protocol, different from the first protocol and further wherein the second protocol is unknown, and wherein the microcode engine is configured to determine, based upon a first portion of the second data stream, the second protocol of the multimedia data of the second data stream.